Coding for the deep-space optical channel

Bruce Moision
Jon Hamkins

331 Section Seminar

October 9, 2001

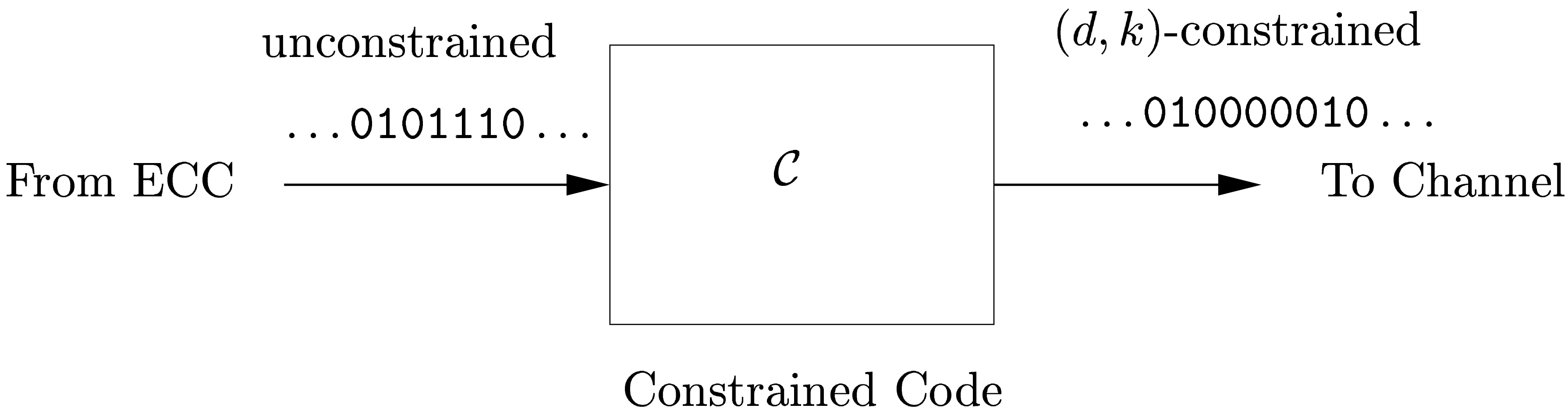
Dead-time constraint

- Q-switched lasers have been proposed for use on the deep-space optical channel as they can confine a high-energy pulse to a narrow slot time T_s. However, the pulses must be separated by a 'dead-time' T_d during which the laser recharges. Assuming On-Off-Keying, this translates into the constraint that 1's be separated by $d = (T_d/T_s)$ 0's.

$$\ldots 000010000000100001000010000000 \ldots$$

- Also typically impose maximum of k 0's, or non-pulsed slots, for timing recovery.
- Refer to this as a (d, k)-constraint, i.e. 1's must be separated by at least d and at most k 0's.

Constrained Code



$$\mathcal{C} : \{0, 1\}^p \to \{0, 1\}^q$$

- What are the achievable rates, $R_{\mathcal{C}} = p/q$, of such a code?
- What are the tradeoffs? E.g. complexity/throughput/transmitted energy/performance?
- What is the performance in a larger coding scheme?

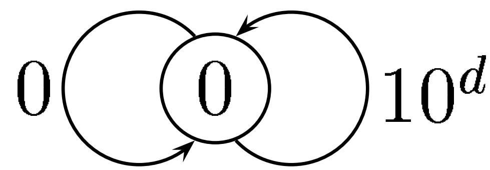
Outline

1. **Some constrained codes, their rates, complexity of implementation, and performance.**

2. **How these codes perform in a serially-concatenated coding scheme, and how their performance compares with baseline.**

Achievable Rates

Describe allowable sequences as paths on a labelled graph.
Consider rates relative to (d, ∞), and take k as a design parameter.



Capacity,

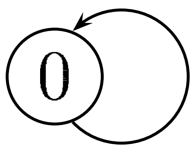
$$C(d) \stackrel{\text{def}}{=} \lim_{n\to\infty} \frac{1}{n} \log |\text{words of length } n \text{ in the } (d, \infty) \text{ system}|$$

upper bounds achievable rates. For large d, we have *[Shannon, 48],[Khandekar, McEliece, 99]*.

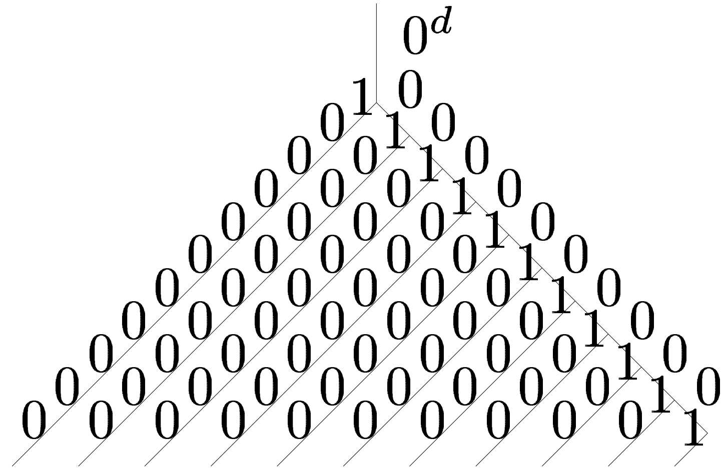
$$C(d) \approx \frac{1}{T_s \ln(2)} \frac{W(d+1)}{d+1} \text{bits/s}$$

where $W(z)$ is the *productlog* function which gives the solution for w in $z = we^w$.

Baseline: Pulse-Position-Modulation (PPM)



$$\begin{array}{c} 10^{M-1}0^d \\ 010^{M-2}0^d \\ \vdots \\ 0^{M-1}10^d \end{array}$$



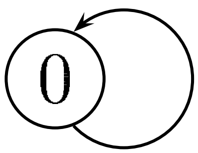
$$R_{\mathrm{PPM}}(d, M) = \frac{1}{T_s} \frac{\log_2(M)}{M+d} \text{ bits/s}$$

Choosing M to maximize the rate,

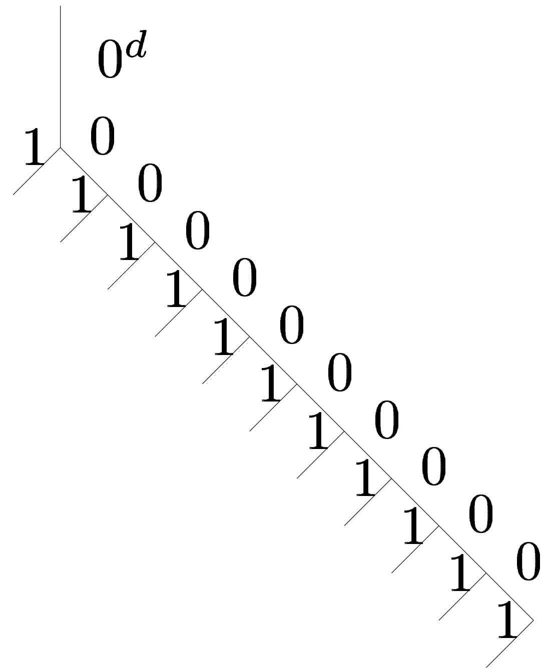
$$R_{\mathrm{PPM}}(d) = \frac{1}{T_s \ln(2)} \frac{W(d/e)}{d} \text{ bits/s}$$

One can show that $R_{\mathrm{PPM}}(d)/C(d) \to_{d\to\infty} 1$, although significant throughput gains are possible for small to moderate d.

Truncated-Pulse-Position-Modulation (TPPM)



$$\begin{array}{c} 10^d \\ 010^d \\ \vdots \\ 0^{M-1}10^d \end{array}$$



TPPM has a maximum *average* rate

$$R_{\mathrm{TPPM}}(d) = \frac{2}{T_s \ln(2)} \frac{W(\frac{2d+1}{e})}{2d+1} \text{ bits/s}$$

$R_{\mathrm{TPPM}}(d) > R_{\mathrm{PPM}}(d)$, hence $R_{\mathrm{TPPM}}(d)/C(d) \to_{d\to\infty} 1$. However, variable rate mapping leads to implementation problems.

Synchronous-Truncated-Pulse-Position-Modulation (STPPM)

Desire gains of variable-length mapping without implementation issues.

- Allow variable-length codewords, but constrain mapping to be synchronous, i.e.,

  $$\mathcal{C} : \{0,1\}^{mp} \to \{0,1\}^{mq}, m = 1, 2, \ldots$$

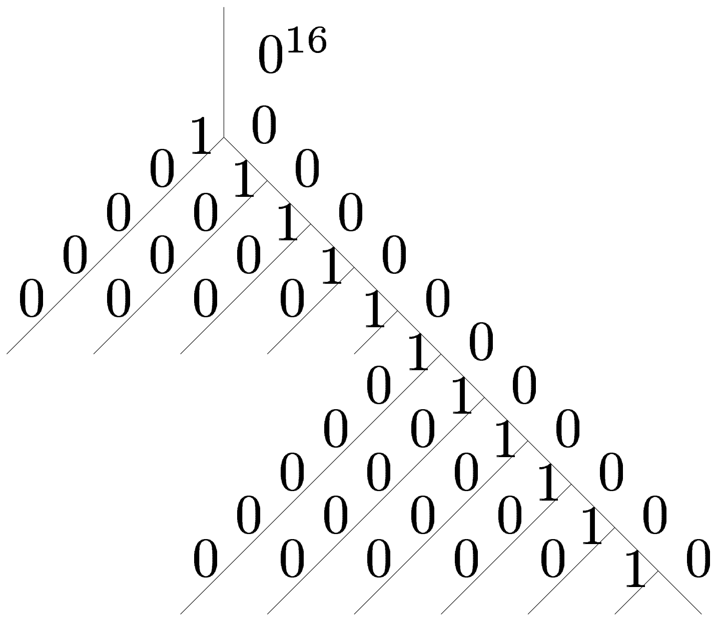
  hence rate is fixed, p/q.

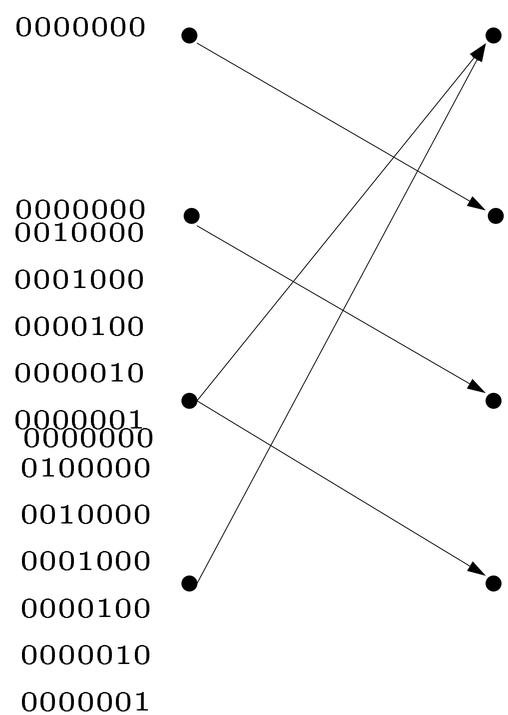
- Choosing $p/q \leq C(d)$, construct a set of prefix-free codewords that can be freely concatenated, with lengths a multiple of q, such that they satisfy the Kraft (In)Equality,

  $$\sum_{c \in \mathcal{C}} 2^{|c|p/q} = 1$$

- Such a set can be used to construct a synchronous code.

STPPM, $d = 16$





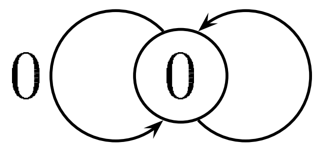
Mappings $3/21, 4/28$, but implemented via fixed rate $1/7$ trellis. Maximum-Likelihood decoder can use variable out-degree trellis with no modifications. Maximum-A-Posteriori decoder may also use the trellis, using, e.g. BCJR algorithm, with some modifications.

STPPM: Variable-out-degree implementation

d	q	fixed-out-degree $\leq$ |States| $\leq$		STPPM|States/Edges| variable-outdegree	fixed-out-degree
16	7	4	16	4/14	10/20
32	10	8	64	8/51	36/72
64	17	12	160	6/41	35/70
128	28	20	448	7	51/102
	27	25	688	10/146	130/260
256	48	32	2448	8	112/224
	46	44	2940	12/301	270/520
512	81	73	8887	11/385	342/684

Nonpulsed-Position-Modulation (NPM)

Allow non-pulsed frames, i.e, $k = \infty$. Presented as a $(\lceil d/M \rceil, \infty)$ constraint over the $M+1$-ary alphabet $\{0, 1, \ldots, M\}$, where each element is a PPM symbol (0 denotes an M-ary non-pulsed frame).



$$
\begin{gathered}
10^{\lceil d/M \rceil} \\
20^{\lceil d/M \rceil} \\
\vdots \\
M0^{\lceil d/M \rceil}
\end{gathered}
$$

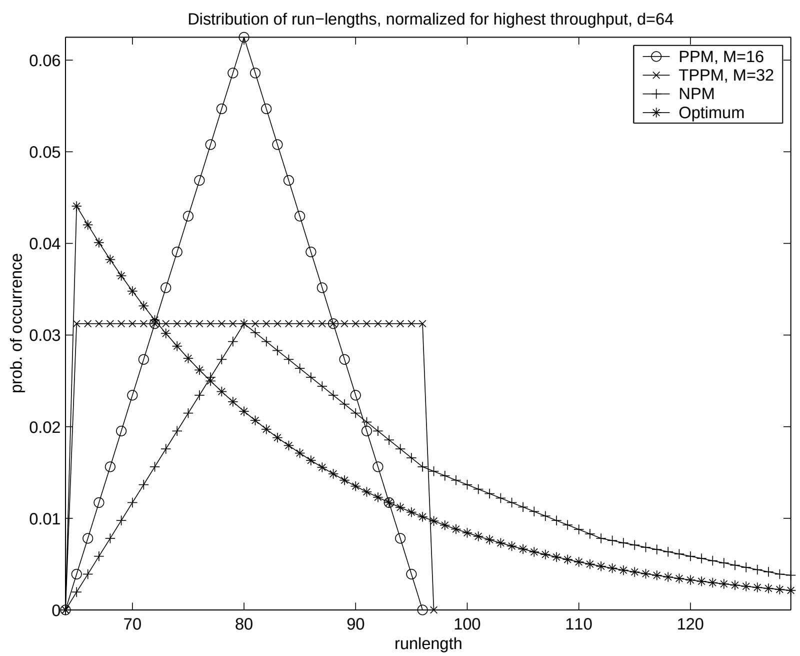
For large d/M,

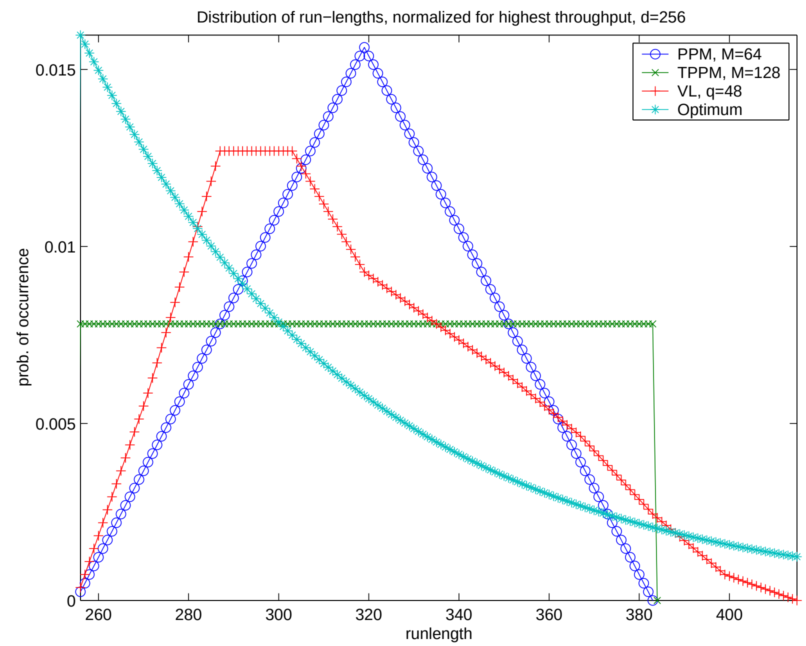
$$
C(M, d) \approx \frac{1}{T_s \ln(2)} \frac{W(d+M)}{d+M} \text{bits/s}
$$

For some pairs (M, d), there exist capacity-achieving, minimal complexity codes into the $(\lceil d/M \rceil, \infty)$ constraint *[S.W. McLaughlin, 97]*

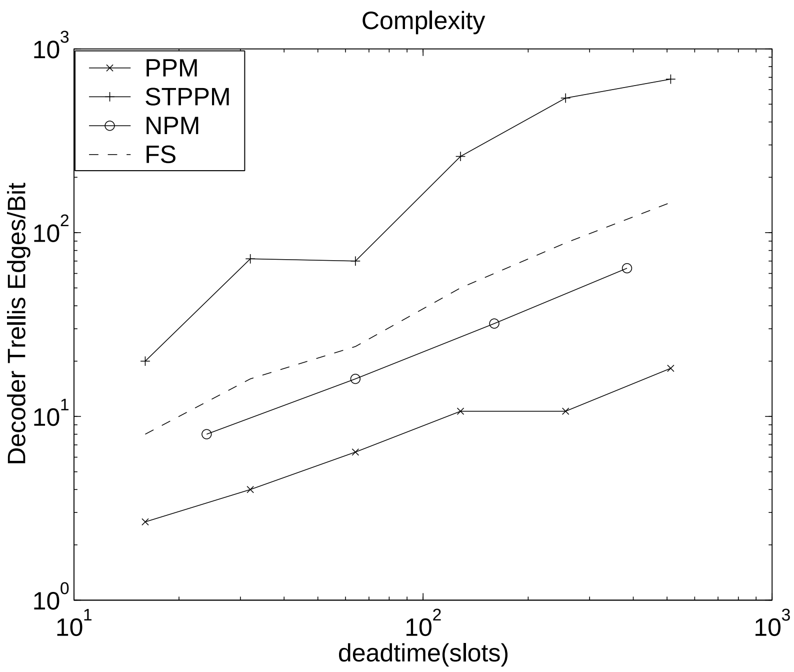
Run-length distributions

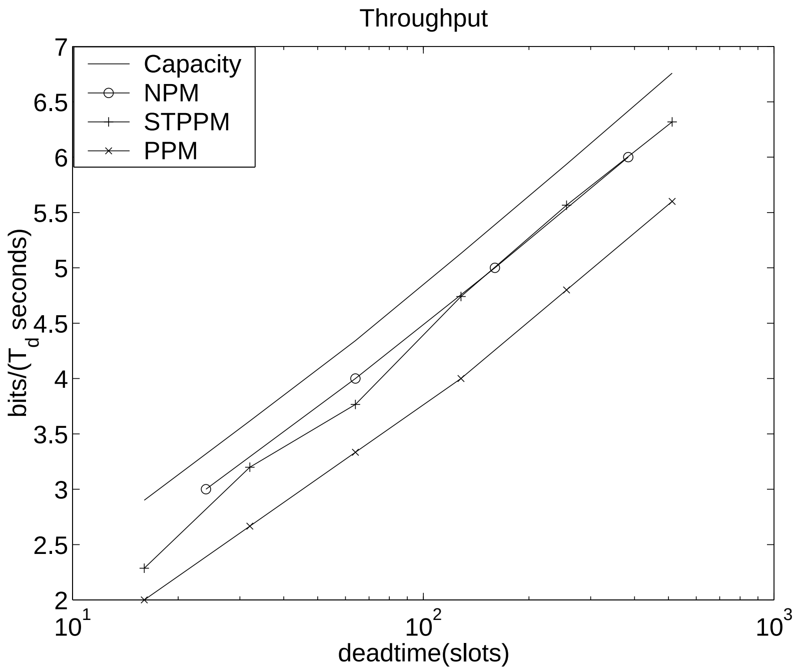
A sequence in a (d, ∞)-constrained code may be uniquely parsed into phrases $0^j 1$ where $j \geq d$. The distribution of these *runlengths* will affect timing-recovery.



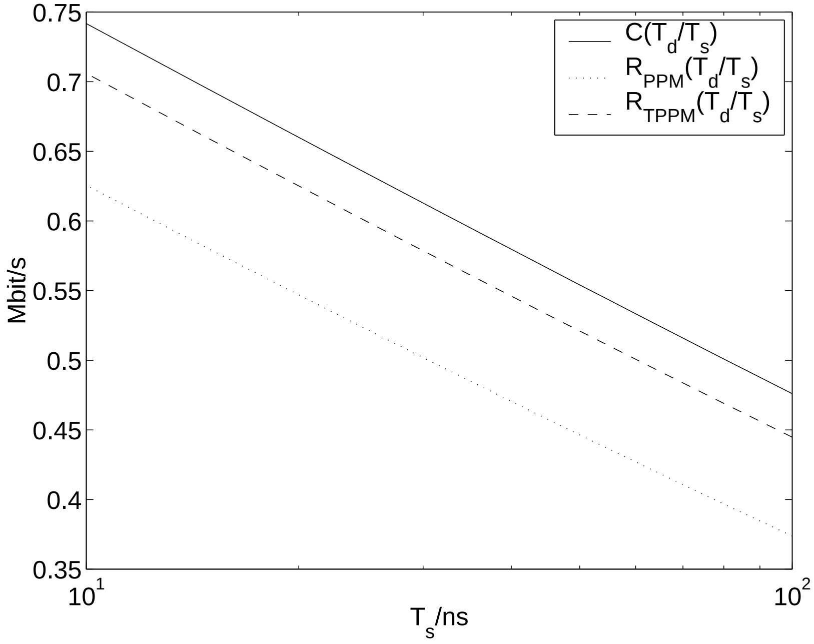


Throughput, Complexity



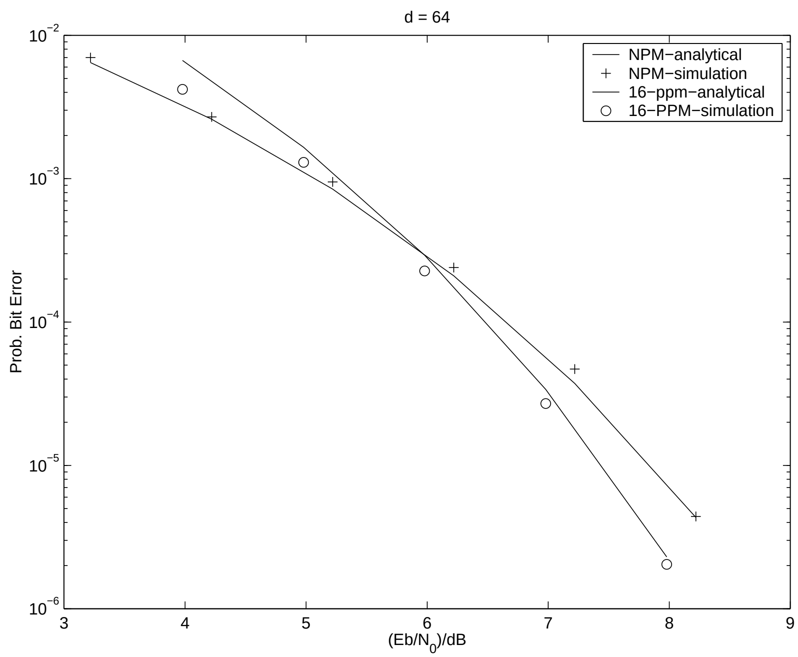


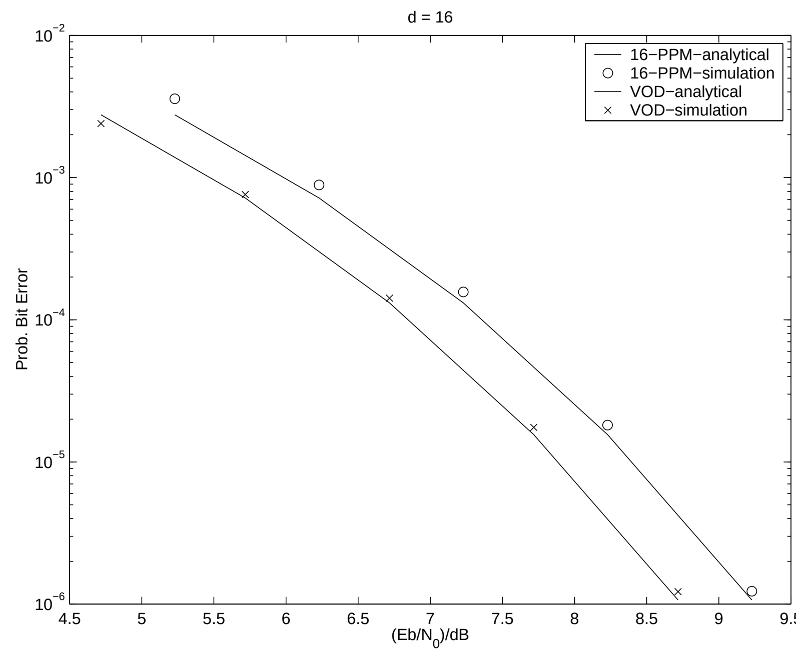
Throughput for fixed T_d



$$C(T_d/T_s) \approx \frac{1}{\ln(2)} \frac{W(T_d/T_s)}{T_d} \text{bits/s}$$

Performance





BCJR: edge-defined α's

We use a novel(?) implementation of the BCJR, defining

$$\alpha_k(e) \stackrel{\text{def}}{=} P\{e_k = e,\ \text{observation}\ \},$$

where e_k is the *edge* traversed at time instant k, as opposed to

$$\alpha_k(s) \stackrel{\text{def}}{=} P\{s_k = s,\ \text{observation}\ \}, \textit{[BCJR, 74]}$$

where s_k is the *state* traversed at time instant k.

For a fixed-out-degree code, the cost per iteration is approximately

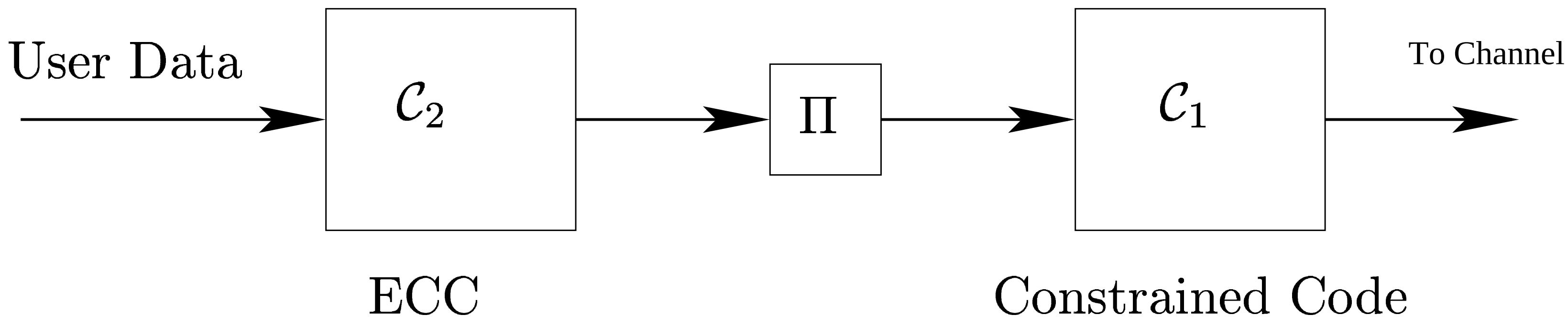
$$3N|\mathcal{E}| \text{ multiplications, } 3N|\mathcal{E}| - 2N|\mathcal{V}| \text{ additions,}$$

where $\mathcal{E}$ is the set of edges, $\mathcal{V}$ is the set of states. State-based algorithms have approximate cost per iteration

$$4N|\mathcal{E}| \text{ multiplications, } 3N|\mathcal{E}| - 2N|\mathcal{V}| \text{ additions.}$$

Gain occurs only for events that are functions of edges.

Concatenating the constrained code



- Constrained code will be concatenated with an outer Error Correcting Code (ECC).
- Baseline is Reed Solomon concatenated with PPM ($RS(M-1,k) \leftarrow$MPPM).
- Other orders of concatenation, e.g. those considered for magnetic or optical storage, are inappropriate here.

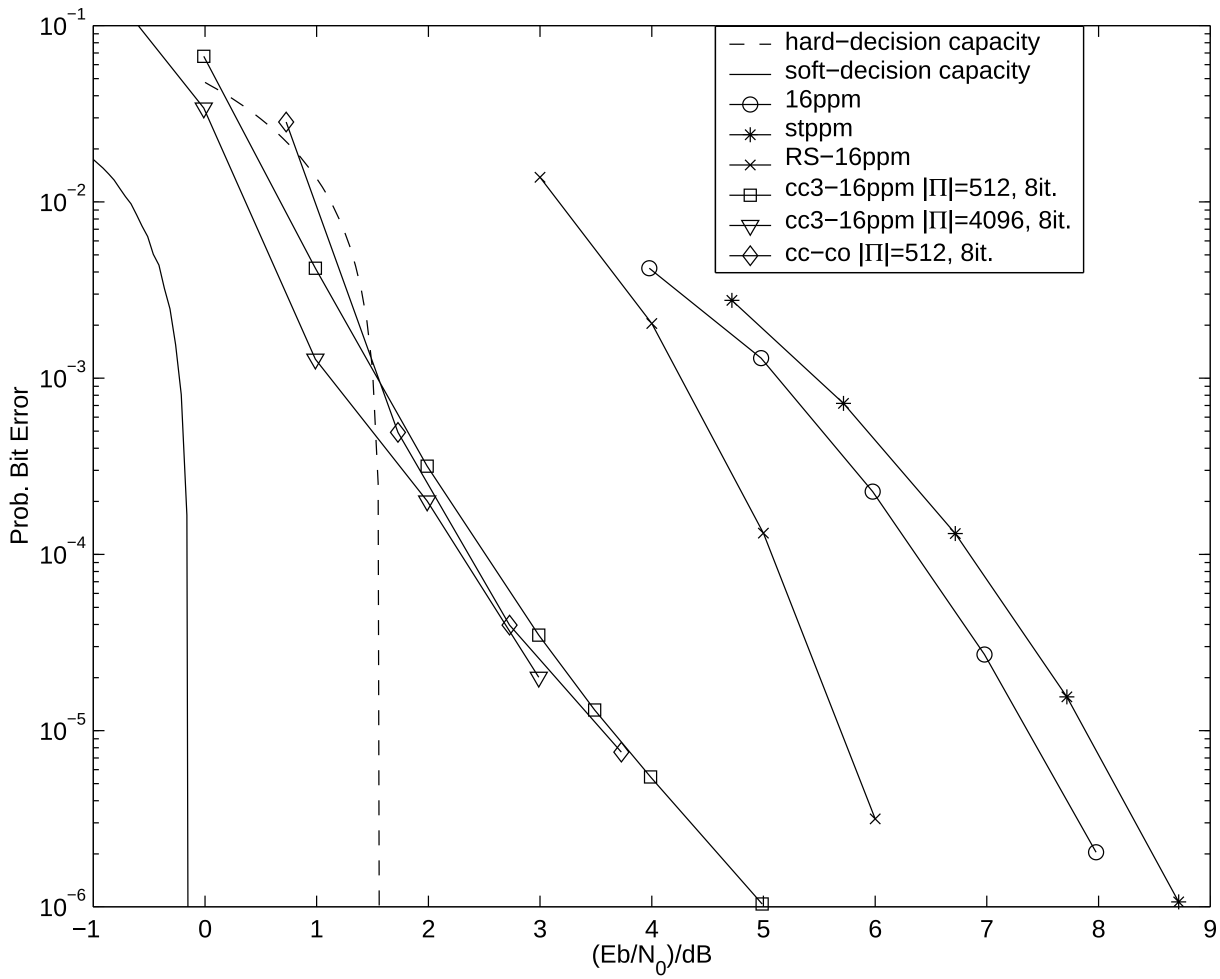
Prior Work

- PCCC←PPM *[Hamkins, 99]* on AWGN, Webb, Webb+Gaussian channel models.
- PCCC↔PPM *[Peleg, Shamai, 00]* Included PPM in iterations on *discrete-time memoryless rayleigh fading channel*. Illustrated performance 1–2 dB from capacity. PPM introduced to yield distribution close to capacity achieving.

Proposed system

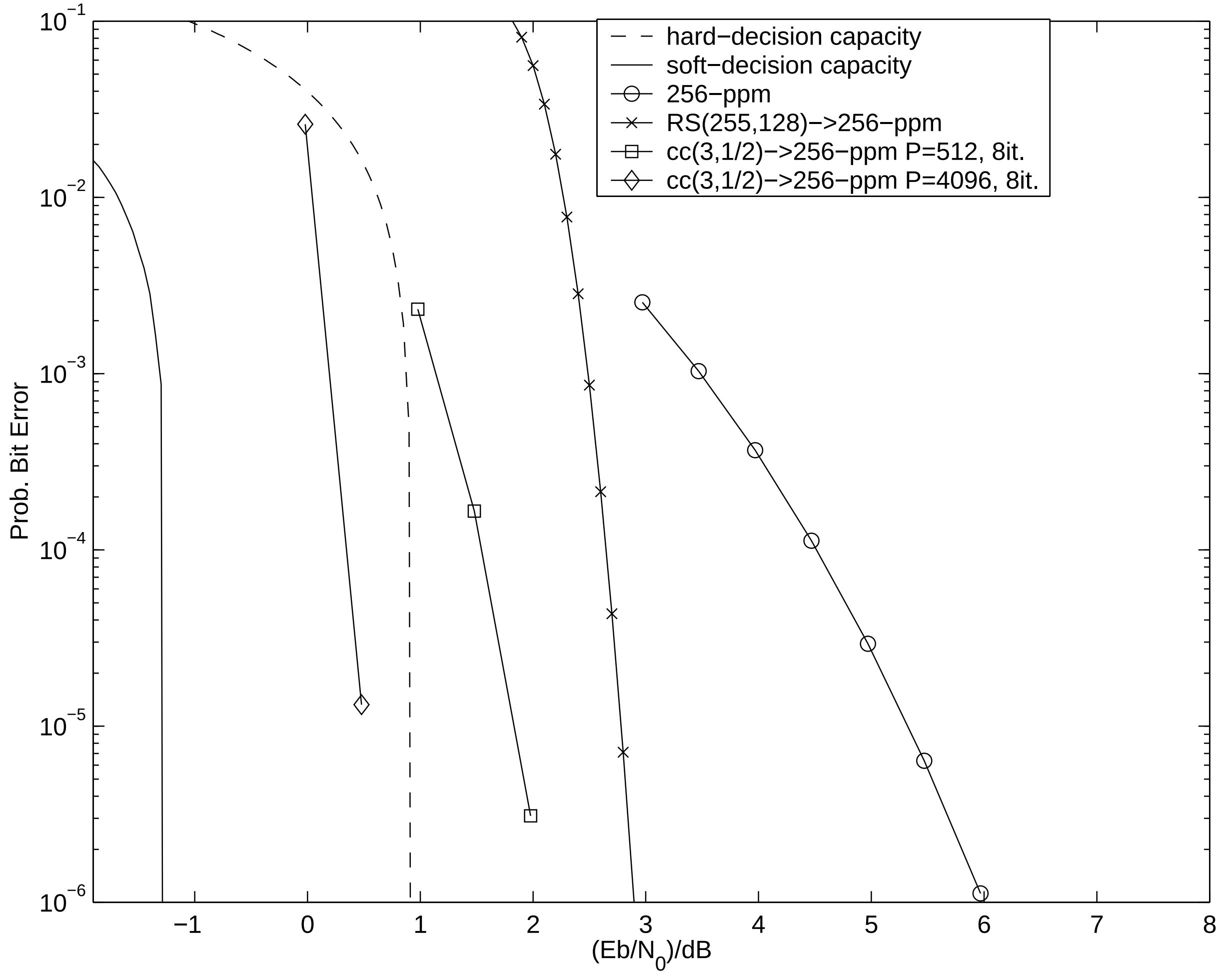
- We illustrate that the system $CC(3, 1/2)$ ↔PPM, or $CC(3, 1/2)$ ↔STPPM, where CC$(3, 1/2)$ is a 4-state convolutional code, provides substantial gains over RS$(M-1, k)$ ←PPM, moderate gains over PCCC←PPM, and small losses relative to PCCC↔PPM.

$d = 16$



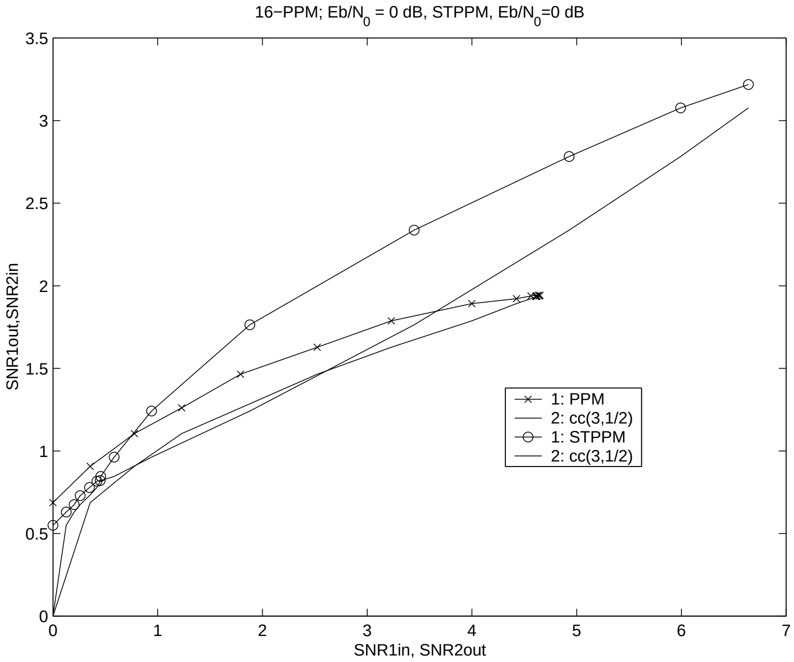
hard−decision capacity
soft−decision capacity
16ppm
stppm
RS−16ppm
cc3−16ppm |Π|=512, 8it.
cc3−16ppm |Π|=4096, 8it.
cc−co |Π|=512, 8it.
Prob. Bit Error
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
−1
0
1
2
3
4
5
6
7
8
9
(Eb/N_0)/dB

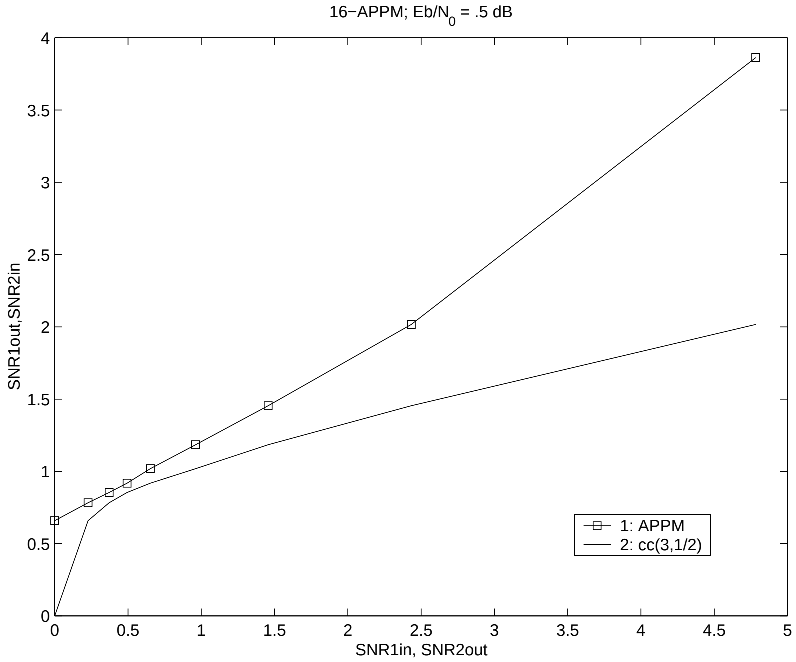
$d = 256$



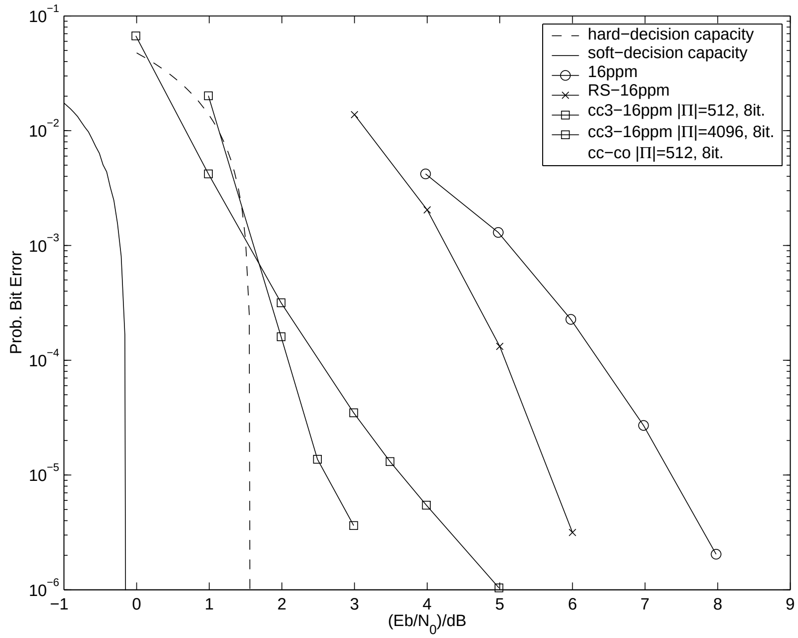
hard−decision capacity
soft−decision capacity
256−ppm
RS(255,128)−>256−ppm
cc(3,1/2)−>256−ppm P=512, 8it.
cc(3,1/2)−>256−ppm P=4096, 8it.
Prob. Bit Error
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
−1
0
1
2
3
4
5
6
7
8
(Eb/N_0)/dB

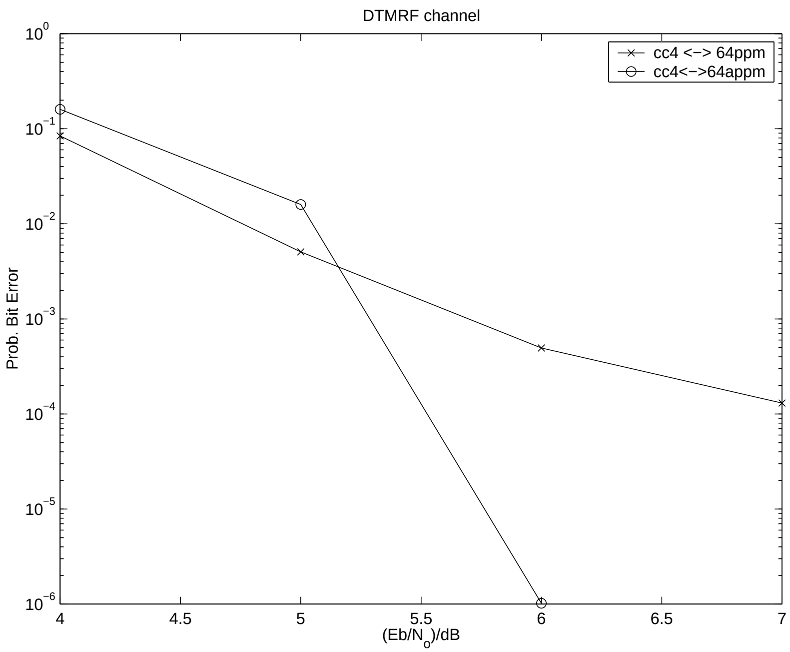
Density Evolution Analysis





Accumulate-PPM performance





Conclusions